REMARKS

Claims 1-36 were previously pending in this application. Claims 1, 25 and 26 are amended. Claims 15-24 and 27-36 are canceled. No claims are added. Claims 1-14, 25 and 26 remain pending.

35 U.S.C. § 102 Rejections

Claims 1-5, 7-9, 15-22, 25-31 and 33

Claims 1-5, 7-9, 15-22, 25-31 and 33 stand rejected under 35 U.S.C. 102(b) as being anticipated by Viroli and Natali ("Parametric Polymorphism in Java through the Homogeneous Translation LM: Gathering Type Descriptors at Load-Time", DEIS Technical Report No. DEIS-LIA-00-001, 2000) (hereinafter "Viroli"). Applicant respectfully traverses the rejection.

Generally, the presently claimed subject matter is concerned with providing a typing context for execution of operations that involve parametric polymorphism. The typing context for each polymorphic expression is characterized using a dynamically allocatable runtime type descriptor (RTD) that records the exact type of an associated generic type.

The RTDs associated with a given open-type expression are accessible via a typing context handle (TCH) that records the typing context in which the expression is executing. The dynamic allocation of RTDs allows for an arbitrary number of RTDs and supports multiple machine code forms for the intermediate language code. In contrast to previous approaches that employ statically allocated RTDs, dynamically allocated RTDs need not be allocated for parameterized objects defined in unexecuted code.

Viroli describes the introduction of parametric polymorphism in Java with a homogeneous translation approach. Viroli discloses a homogeneous translation in which runtime information about instantiation of type parameters is carried and thus allowing integration of parameterized types with Java typing. The translation technique identified defers management of type information until load time. This provides stated advantages over previous heterogeneous translation approaches.

Claim 1

Claim 1 has been amended and now recites a "computer program product encoding a computer program for executing on a computer system a computer process for dynamically generating typing context data associated with a

typing-context-relevant-code-point being executed within a typing context in a dynamic execution environment." The computer process comprises steps of:

(1) "encountering the typing-context-relevant-code-point in the typing context during execution of the program;"

(2) "identifying a typing context handle associated with the typing context, the typing context handle referencing a typing context data structure associated with the typing context;"

(3) "computing the typing context data associated with the typing-context-relevant-code-point;"

(4) "dynamically allocating a field in the typing context data structure associated with the typing-context-relevant-code-point;" and

(5) "recording the typing context data in the field of the typing context data structure." (Amendment emphasized).

Claim 1 has been amended to clarity that slots associated with the Typing Context (TC) data structure are dynamically allocated as new "open type expressions" are discovered and required. Such a dynamic allocation of slots for "open type expressions" allows for an arbitrary number of "open type expressions." In contrast to the cited reference and to other previous

approaches that employ statically allocated RTDs (Run-time Type Descriptors) and/or statically allocated slots for "open type expressions," dynamically allocating RTDs and slots means that RTDs and slots need not be allocated for parameterized objects defined in unexecuted code.

Such dynamic allocation in a data structure is not disclosed or anticipated by the cited reference (Viroli). A somewhat corresponding data structure in Viroli is called "$TDManager (Type Descriptors Manager). . . . Methods and fields of $TDManager will be declared static." (p. 8, paragraph 4.1)

Accordingly, claim 1 is allowable over the cited reference and the rejection thereof should be withdrawn.

Claims 2-5 and 7-9 depend from claim 1 and are allowable at least by virtue of that dependency. Therefore, the rejection of these claims should be withdrawn.

Claims 15 -22 have been canceled, thus rendering the rejection thereof moot.

Claim 25 has been amended to recite "an execution engine for executing parametrically polymorphic code and dynamically generating typing context data associated with a typing-context-relevant-code-point being executed

within a typing context in a dynamic execution environment." The execution engine comprises:

(1) "a read module configured to encounter_ the typing-context-relevant-code-point in the typing context during execution of the program;"

(2) "a handle module configured to identify_ a typing context handle associated with the typing context, the typing context handle referencing a typing context data structure associated with the typing context;"

(3) "a computation module configured to compute_ the typing context data associated with the typing-context-relevant-code-point;"

(4) "an allocation module configured to dynamically allocate a field in the typing context data.structure associated with the typing-context-relevant-code-point;" and

(5) "a recording module configured to record the typing context data in the field of the typing context data structure." (Relevant amendment emphasized).

Similar to claim 1, fields in the typing context data structure are dynamically allocated to allow for conservation of memory space and, to some extent, computational overhead. Unlike the cited references and other methods

which define type descriptor object fields statically, when allocating such fields dynamically, a number of fields does not need to be predetermined and code modules not executed will not have to have memory reserved for them.

Since each and every element of claim 25 is not disclosed or anticipated by the cited reference, claim 25 is allowable over Viroli and the Section 102 rejection of claim 25 should be withdrawn.

Claim 26 has been amended to recite "a method of dynamically generating typing context data associated with a typing-context-relevant-code-point being executed within a typing context in a dynamic execution environment." The method comprises the steps of:

(1) "encountering the typing-context-relevant-code-point in the typing context during execution of the program;"

(2) "identifying a typing context handle associated with the typing context, the typing context handle referencing a typing context data structure associated with the typing context;"

(3) computing the typing context data associated with the typing-context-relevant-code-point;"

(4) "dynamically allocating a field in the typing context data structure associated with the typing-context-relevant-code-point;" and

(5) "recording the typing context data in the field of the typing context data structure."

Similar to claims 1 and 25 discussed above, the dynamic allocation recited in claim 26 is not disclosed or anticipated by the cited references. Accordingly, claim 26 is allowable over Viroli and the rejection of claim 26 should be withdrawn.

Claims 27-31 and 33 have been canceled, thus rendering the rejection thereof moot.

35 U.S.C. § 103 Rejections

Claims 6, 10-14, 23, 24, 32 and 34-36

Claims 6, 10-14, 23, 24, 32 and 34-36 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Viroli in view of U.S. Patent No. 5,093,914 issued to Coplien et al. (hereinafter "Coplien"). Applicant respectfully traverses the rejection.

Claims 6 and 10-14 depend from claim 1 and are allowable at least by virtue of that dependency for the reasons stated in the response to the rejection of claim 1, above.

Accordingly, these claims are allowable over the cited combination of references for at least this reason and the rejection of these claims should be withdrawn.

Claims 23, 24, 32 and 34-36 have been canceled, thus rendering the rejection thereof moot.

CONCLUSION

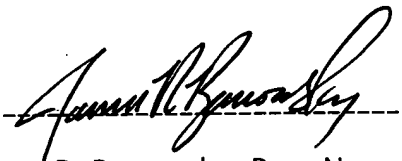
Accordingly, in view of the above amendment and remarks it is submitted that the claims are patentably distinct over the prior art and that all the rejections to the claims have been overcome. Reconsideration and reexamination of the above Application is requested. Based on the foregoing, Applicants respectfully requests that the pending claims be allowed, and that a timely Notice of Allowance be issued in this case. If the Examiner believes, after this amendment, that the application is not in condition for allowance, the Examiner is requested to call the Applicant's attorney at the telephone number listed below.

If this response is not considered timely filed and if a request for an extension of time is otherwise absent, Applicants hereby request any necessary extension of time. If there is a fee occasioned by this response, including an extension fee that is not covered by an enclosed check please charge any deficiency to Deposit Account No. 50-0463.

Respectfully submitted,

Microsoft Corporation

Date: 6-2-05

By: 

James R. Banowsky, Reg. No.: 37,773
Attorney for Applicants
Direct telephone (425) 705-3539
Microsoft Corporation
One Microsoft Way
Redmond WA 98052-6399